Decision trees as a computational model for mental heuristics in human decision-making: a study of poker duels

Background: The need for an intuitive computational model of cognition

Neural networks are one of the most popular machine learning models because of their ability to capture properties in data that may not be perceivable by the eye. However, neural networks face several disadvantages as models of human cognition.

Firstly, while they may seem to be inspired by how the brain works, this relation is only superficial. This is evidenced by, for example, the phenomenon of adversarial machine learning. Computer vision models can correctly classify images with high precision, but if those images are perturbed with changes imperceptible to the human eye, they are completely misclassified (Tramer et al., 2018). Figure 1 illustrates this (Goodfellow et al., 2014). To the left, a model correctly classifies an image as a panda, but after adding an insignificant perturbation, the image is classified as a gibbon with almost 100% confidence (a lot higher than the original confidence for panda).

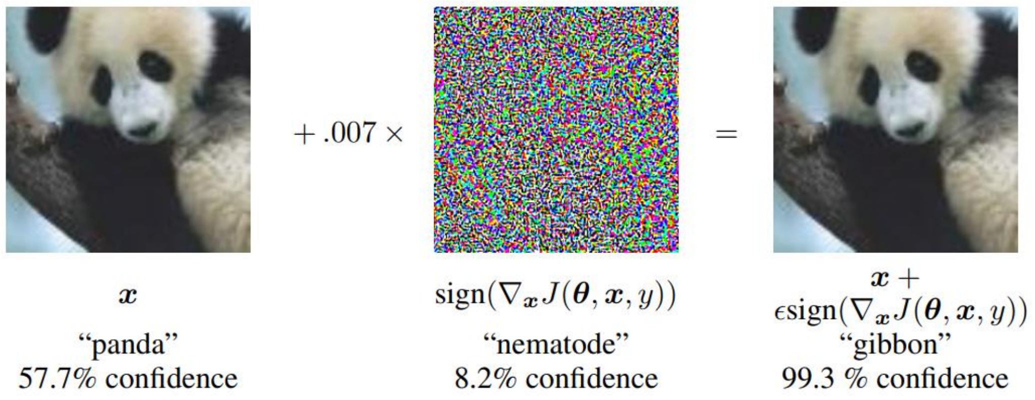

$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1. Adversarial image generation using the Fast Gradient Sign Method (FGSM).

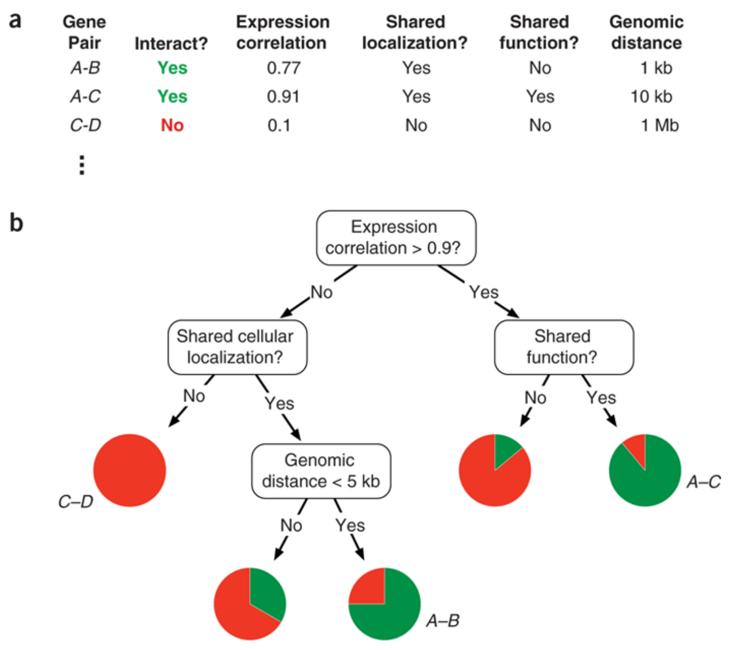
Secondly, neural networks are a 'black box'. They may be incredibly effective at predicting future events, and have the powerful ability to imitate language and vision (to a limited extent), but we do not know how they do it in relation to the input features they are given. That is, they do not allow us to study the inner mechanisms that lead to neural networks producing those results. Because of this, it is not possible to claim that NNs are a model of human cognition, because producing the same results does not mean that the process through which those results are obtained is the same, or even slightly similar. It is currently not possible to draw relevant parallels between NNs and the human brain and cognition.

The problem with this is clear: there is a need for a simpler, more intuitive machine learning algorithm that allows us to both make good predictions and look at how the algorithm is making those predictions. If we can look at the inner computations being performed, we can tell with more certainty whether the algorithm is actually modeling human cognition. If it is, it would allow us to study cognition from a computational view in a much more approachable manner. A natural candidate for this are decision trees.

Decision trees as a model of human cognition

Decision trees are classifiers that predict class labels for data items. They consist on a series of ordered nodes, which perform an evaluation on the features of the data in order to reach a conclusion about its label. They are able to handle discrete, continuous, and Boolean data without problem. For example, oncologists use them to classify tumors based on biopsies, patient records, and medical evaluations.

Decision trees also have a huge benefit in comparison to neural networks: they allow you to see the process they follow to reach a conclusion. Figure 2 (Kingsford & Salzberg, 2008) shows a decision tree used to classify whether a pair of genes will interact based on other given features. As it can be seen, each node consists on a yes or no question, and depending on the answer, you move down the tree until you reach a leaf, which is a node with no further path down. Pie charts at the bottom represent the proportion of genes that interact and do not interact once the data has been divided that way. For example, if you reach a node which is mostly green, it would be fair to assume that most pairs of genes which end up in that leaf of the decision tree will interact.



Gene Pair	Interact?	Expression correlation	Shared localization?	Shared function?	Genomic distance
A-B	Yes	0.77	Yes	No	1 kb
A-C	Yes	0.91	Yes	Yes	10 kb
C-D	No	0.1	No	No	1 Mb
⋮					

Figure 2. Gene-pair interaction classification.

It is clear how decision trees allow us to easily visualize the decision-making process. I believe this makes them an ideal candidate to study human cognition. But how are decision trees related to the way human beings decide? It is a well-known fact that human decision-making uses a vast amount of mental heuristics to make decisions all of the time. These allow us to avoid mathematically and rationally evaluating every single alternative to obtain results, which may be equally good, or even better, when we are under pressure. It seems plausible that, when we are forced to make quick decisions, we use rules of thumb and mentally answer quick questions that let us choose an alternative. Decision trees could potentially be used to model human mental heuristics.

Texas Hold’em Poker as a sandbox to study decision-making

Studying this in real-world decisions would be incredibly hard. Luckily, card games provide a playground for humans to put their decision-making abilities to test. More specifically, Texas Hold’em Poker puts players to compete against each other, in an environment with uncertain and imperfect information, with the goal of winning money. It is a very popular game, with thousands of players voluntarily opting into it every single day, both online and in-person.

The rules of poker, in brief terms, are as follows. Every player is dealt 2 cards (pre-flop), everyone gets a chance to bet money. Then, cards are dealt in the board, so that you can match them with your hand to get good card combinations. In the end, the last player to stand in the bets, or the best hand, takes all the money from that round. Since you do not know what cards your opponent has, the game does not have perfect information, so you have to calculate risk, consider the possibility that your opponents may be bluffing, and think about how to best maximize your expected earning with every single decision you make. Under such pressure to make decisions quickly with the goal of earning money, it is reasonable to believe people would employ mental heuristics. This mental heuristic could be a decision tree, since people need good rules of thumb to make good choices with the available information, and decision trees thrive doing that.

However, this leaves one question open: do people use the same decision tree every time they play poker? Some professional poker players say that “you don’t play the cards, you play the players”. This implies that choices that may be correct against one player could be incorrect against another. Some computer models have validated this idea by implementing opponent modeling (Billings et al., 1998; Yan et al., 2020). These models, instead of just making decisions based on the current game state, model the decision making process of the opponent, so that they can exploit their specific weaknesses.

However, it is one thing to say that opponent modeling is a strategy that works, and a completely different thing to claim that professional poker players are actually performing the mental operation of modeling their opponents. So how can we tell if professional poker players are actually modeling the way their opponents play? It could perfectly be the case that they think they are playing the player and not the cards, but that they really play same against all of their opponents without realizing it. I propose to model the decision-making process of professional poker players with decision trees to study this. Comparing the resulting tree structure from training the algorithm in relevant datasets could give us an insight into how professional poker players play the game.

Hypothesis and Data Collection

I hypothesize that professional poker players create mental models of their opponents, and then use these models as a heuristic to help their decision-making process. We will use decision trees to test this hypothesis. If my hypothesis is true, then different decision trees should be produced when analyzing the data of a single player versus two different opponents in duel (1v1) games.

We will put this hypothesis to test by analyzing two poker videos from the show High Stakes Duel, which can be found in the PokerGo streaming platform (I subscribed for a month to get access to the videos). The first video features Phil Hellmuth playing a duel versus Daniel Negreanu, and the second video consists on Phil Hellmuth playing a duel versus Tom Dwan. These three players are professionals who have been playing the game for years. They have faced each other multiple times, so they are not new to each other's playstyle. In the High Stakes Duel show, they played several duels, and this is the second duel out of a larger series. The prize for the match was 200,000 dollars in both cases, so all players have a huge incentive to try their hardest to win.

I collected data on several features of the game: Game state, dealt hand, board cards shown, pot size, amount of money to call, bank at the time of making the decision (in number of big blinds), opponent's bank at the time of making the decision (in number of big blinds), and of course, the decision made. All of this data was collected on Phil Hellmuth, since he is the player present in both duels. In total, I collected data for over 140 decisions made by Phil Hellmuth.

The mentioned collected data, on itself, is useless, because a decision tree cannot capture the rules of poker. That is, it does not understand which hands are good and which hands are bad, so we need to process our data to give it meaning. For the pre-flop, which is the state of the game where all the cards available are the two cards only you can see, cards were given a score according to the following formula.

$$hand\ sore = 1 - \frac{hand\ percentile}{100}$$

This metric gives a score of 1 to the best possible hand (AA), a score of 0 to the worst possible hand (72o), and everything else is in the middle. The hand percentile variable refers to the position of the hand in comparison to the total amount of starting hands possible.

For the other stages of the game (flop, turn, and river), I had to use a python package called Treys (*Treys*, n.d.). Treys allows you to input the cards in your hand and the cards in the board, and gives an evaluation of the hand you have. The strongest possible hand, a Royal Flush, is given a strength (which I will refer to as ranking) of 1, and the worst possible hand is given a ranking of 7462.

As I mentioned, data on the decision made by the player given a certain game state was also collected. However, not all decisions in poker are the same. It is important to understand some of the nuance of the game to know what type of decisions are relevant, and how they are relevant. More specifically, I analyzed two different situations in which players need to make decisions.

Situation 1: Stay vs Leave

This is the most basic of scenarios, since this decision needs to be made every single time. Players have several options: Check (passing on the opportunity to bet), Bet (to wager an initial amount), Raise (to bet more than the minimum amount or more than a bet to you), Call (matching the bet of a player), and Fold (giving up the hand) (*Poker Terms - Online Poker Lingo Dictionary*, 2008).

Checking, calling, betting or raising were treated as **staying**, while folding was treated as **leaving.** Treating decisions as binary allows us to get a simpler first approach to the problem.

This situation was analyzed in separate scenarios: pre-flop and post-flop (I say post-flop but it really includes the flop). Since the pre-flop has only two possible cards, the way of ranking the hands is different. For the pre-flop, I used my proposed metric for hand ranking. For the post-flop, I used Trey's hand ranking algorithm. In total, 4 decision trees are produced for this situation: a pre-flop and a post-flop for Hellmuth against each player.

Situation 2: Check vs Raise

Sometimes, there is no reason whatsoever for a player to fold. When a player is given the opportunity to bet without having to call a previous bet, staying in the round is free, so there is no rational reason for anyone to fold in this scenario. However, players still need to make a different decision: to not bet any money, or to bet some money.

In this situation, calling and checking (they are really the same here) were treated as **checking,** since they maintain the amount of money being bet (0), while betting and raising were treated as **raising**, since they raise the amount of money being bet.

This situation was only analyzed in the post-flop, since there were no situations where it made no sense to fold for any of the pre-flop analyzed hands. Therefore, two decision trees were produced: one for the post-flop of Hellmuth against each player.

Methods

TensorFlow's dataforest package was used to construct the desired decision trees. Since this package only runs on Linux, I had to do this part of the code in a Google Colab instead of a Jupyter Notebook, because Google Colab runs on Linux on the cloud instead of on my own computer. This involves converting our dataframe to a TensorFlow dataset, training the decision tree on the data, and a couple other steps. The package allows us to visualize the tree without needing to build any new methods, which is very convenient. All relevant files (python notebooks and csv files) are attached to the email.

Results

Situation 1.1: Stay vs Leave, pre-flop

Hellmuth vs Negreanu

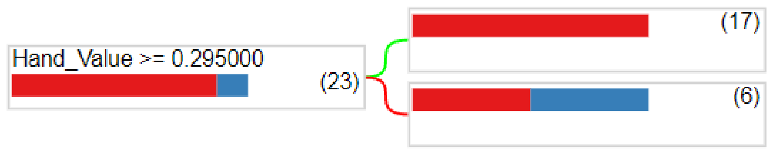


Figure 3. Decision tree for Stay vs Leave, pre-flop, Hellmuth vs Negreanu. In a node, red represents stay, and blue represents leave. Taking the green path after a node means the answer to the question was True, while the red path means it was False.

Hellmuth vs Dwan

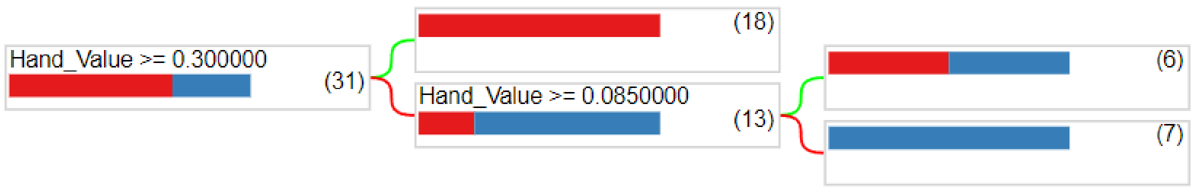


Figure 4. Decision tree for Stay vs Leave, pre-flop, Hellmuth vs Dwan. In the nodes, red represents stay, and blue represents leave. Taking the green path after a node means the answer to the question was True, while the red path means it was False.

As we can see, the pre-flop decision trees for Hellmuth are very similar for both of his duels. This is reflected on the fact that the threshold for the first node is almost the same. However, the second tree is a layer deeper. This could mean that Hellmuth has a pretty straightforward way of playing the pre-flop against Negreanu, but not so much against Dwan. This is compatible with the results we found on the first paper, which indicated that Hellmuth seemed to be willing to fold a wider range of hands against Dwan than against Negreanu. Therefore, while simple, these results are encouraging.

Situation 1.2: Stay vs Leave, post-flop

Hellmuth vs Negreanu

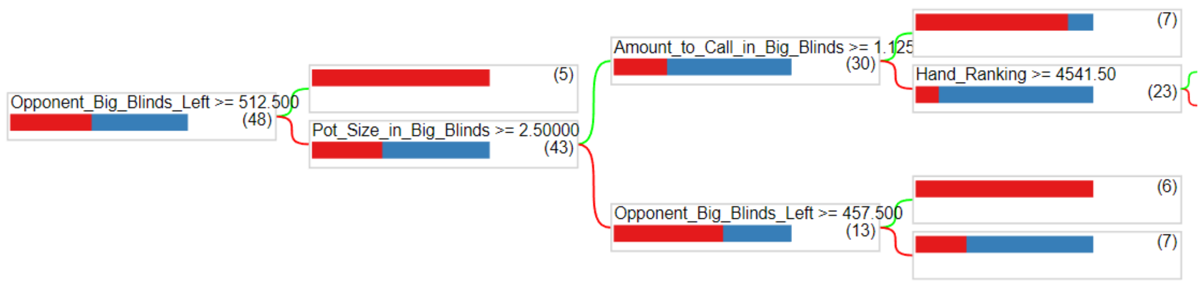

Opponent_Big_Blinds_Left >= 512.500
(48)
(5)
Pot_Size_in_Big_Blinds >= 2.50000
(43)
Amount_to_Call_in_Big_Blinds >= 1.125
(30)
(7)
Hand_Ranking >= 4541.50
(23)
Opponent_Big_Blinds_Left >= 457.500
(13)
(6)
(7)

Figure 5. Decision tree for Stay vs Leave, post-flop, Hellmuth vs Negreanu. In the nodes, red represents stay, and blue represents leave. Taking the green path after a node means the answer to the question was True, while the red path means it was False.

Hellmuth vs Dwan

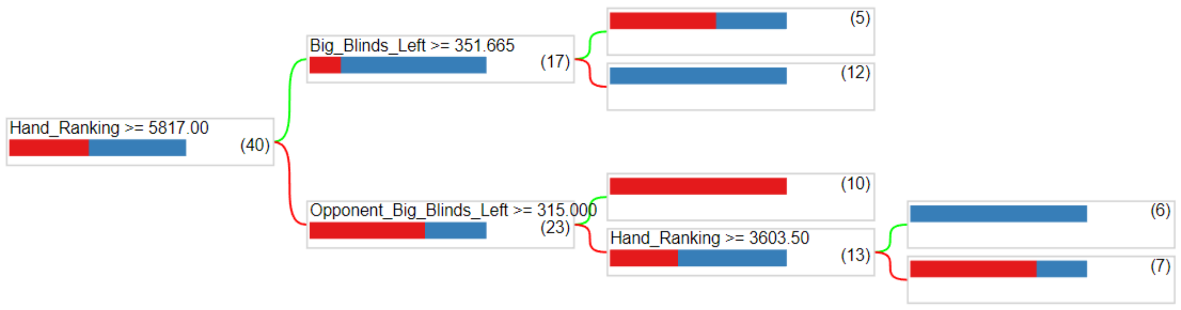


Figure 6. Decision tree for Stay vs Leave, post-flop, Hellmuth vs Dwan. In the nodes, red represents stay, and blue represents leave. Taking the green path after a node means the answer to the question was True, while the red path means it was False.

This is a very interesting result, because a close look at the decision trees allows us to see that they are very different. When Hellmuth plays against Negreanu, he seems to consider the pot size relative to the amount of big blinds the opponent has left a lot more than his hand ranking, which is the first node in the Hellmuth vs Dwan decision tree. This result is also encouraging, because it tells us that what Hellmuth considers to be more important in making a decision depends on who he is playing against, and the decision tree is reflecting that. That is precisely what our hypothesis predicted.

The Hellmuth vs Dwan tree shows us that the algorithm is capturing some curious properties about Hellmuth’s game. For example, if the hand ranking is bad (high), Hellmuth is willing to stay as long as he has enough big blinds left. If he does not have many big blinds left, he is more likely to fold.

Situation 2: Check vs Raise

Hellmuth vs Negreanu

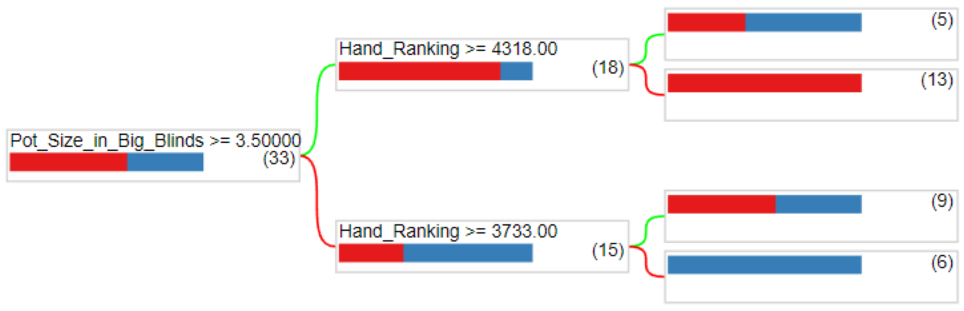


Figure 7. Decision tree for Check vs Raise, post-flop, Hellmuth vs Negreanu. In the nodes, red represents Check, and blue represents Raise. Taking the green path after a node means the answer to the question was True, while the red path means it was False.

Hellmuth vs Dwan

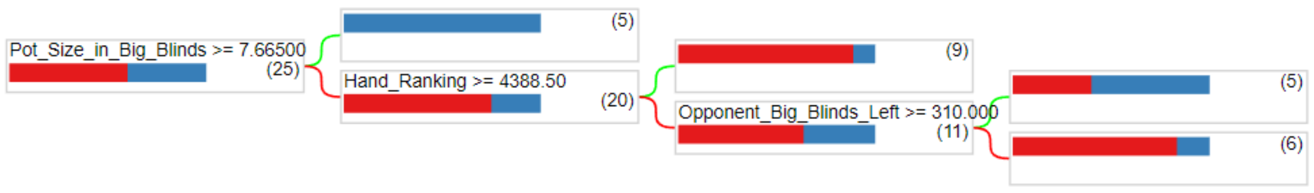

Pot_Size_in_Big_Blinds >= 7.66500
(25)
(5)
Hand_Ranking >= 4388.50
(20)
(9)
Opponent_Big_Blinds_Left >= 310.000
(11)
(5)
(6)

Figure 8. Decision tree for Check vs Raise, post-flop, Hellmuth vs Dwan. In the nodes, red represents Check, and blue represents Raise. Taking the green path after a node means the answer to the question was True, while the red path means it was False.

These results are also encouraging because there are also differences between these two trees. First, while the first node refers to the same feature in both cases, when Hellmuth plays against Negreanu, the threshold is a lot lower.

Secondly, from the second layer of the Hellmuth vs Negreanu tree, we can see that Hellmuth is more likely to raise in either high or low card rankings, but more likely to check in intermediate hand rankings. This is something that is not seen in the Hellmuth vs Dwan tree. The second layer of the Hellmuth vs Dwan tree shows that whenever Hellmuth has a bad hand, he is less willing to bluff Dwan than he is to bluff Negreanu.

In the Hellmuth vs Dwan tree, we see that sometimes a large pot size is enough to convince Hellmuth to raise, which is something that is not seen in his data versus Negreanu.

These trees seem to be capturing some very interesting nuances about the game and the mentality of Hellmuth when playing against each player.

Discussion

Our results are very encouraging because they allow us to confirm our hypothesis that professional poker players play both the cards and the player. Decision trees were considerably different for anything that was not the pre-flop, but even then, there were some differences. Even more, the trees captured some abstract characteristics of the game. For example, the Check vs Raise, Hellmuth vs Negreanu tree, found that Hellmuth was more likely to fold good and bad hands than intermediate hands. This makes sense because if you have high chances of winning, you want to maximize your profits. Similarly, if you have low chances of winning, sometimes you want to bluff, so you raise. However, Hellmuth does not follow the same behavior when playing versus Dwan. In Hellmuth's duel against Dwan, whenever his hand ranking was bad, his preferred option was checking, showing he is less willing to bluff Dwan than he is to bluff Negreanu.

Decision trees have absolutely been overlooked as a model of human cognition. While this project only had about 140 data points, and had a very simple design, it is a very important proof of concept: it is possible to use decision trees to study human decision-making. Further work should consider unifying the existing features so that the relationship between them is captured easily by the tree (for example, combining the amount of money that needs to be called with the current pot size to obtain a ratio).

Data collection can also be enhanced for future developments of the project. The YOLO python package allows for image detection, and it can be used to detect poker cards. However, learning it is not so simple, but it would absolutely be worth the effort. Making the data-collection process automatic would exponentially increase the strength of the results, since they would no longer depend on manual annotations, which are slow and unreliable if one has been doing it for hours.

Works Cited

Billings, D., Papp, D., Schaeffer, J., & Szafron, D. (1998). *Opponent Modeling in Poker*. 7.

Goodfellow, I., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. *ArXiv 1412.6572*.

Kingsford, C., & Salzberg, S. L. (2008). What are decision trees? *Nature Biotechnology*, *26*(9), Article 9. https://doi.org/10.1038/nbt0908-1011

Poker Terms—Online Poker Lingo Dictionary. (2008). https://pokerterms.com/

Tramer, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2018). *ENSEMBLE ADVERSARIAL TRAINING: ATTACKS AND DEFENSES*. 22.

treys: Treys is a pure Python poker hand evaluation library (0.1.8). (n.d.). [Python]. Retrieved December 12, 2022, from https://github.com/ihendley/treys

Yan, X., Xia, L., Yang, J., & Zhao, Q. (2020). Opponent Modeling in Poker Games. *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, 1090–1097. https://doi.org/10.1109/DDCLS49620.2020.9275228